

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-065845

(43)Date of publication of application : 09.03.1999

(51)Int.Cl.

G06F 9/38
G06F 11/28

(21)Application number : 09-231331

(71)Applicant : TOSHIBA CORP

(22)Date of filing : 27.08.1997

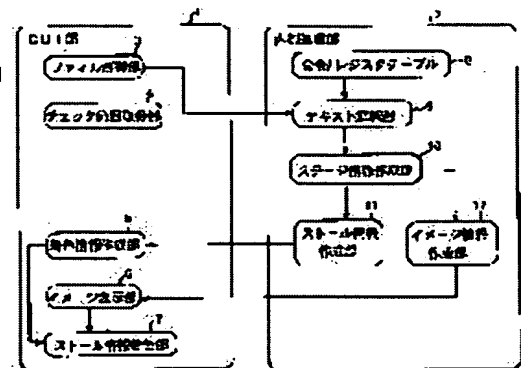
(72)Inventor : SASAKI YOKO
KUNIMATSU ATSUSHI

(54) DEVICE AND METHOD FOR DETECTING AND DISPLAYING STALL

(57)Abstract:

PROBLEM TO BE SOLVED: To make it easy to detect a stall by detecting the stall according to the states of respective stages in the pipeline processing of a source program.

SOLUTION: A file control part 3 reads in the file of the source program. A check range acquisition part 4 acquires a text character string from a file control part 3 according to a stall check range and sends it to an internal processing part 2. A stage information generation part 10 grasps the respective stage states of the pipeline from information of one line obtained from a text interpretation part 9. A stall information generation part 11 checks the stall generation state from the stage states of the pipeline and sends stall information to a GUI part 1. A coloring information generation part 5 generates coloring information for a text and a pipeline image according to the obtained stall information. An image display part 6 displays the source program and the pipeline image information obtained from an image information generation part 12.



LEGAL STATUS

[Date of request for examination]

17.04.2001

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

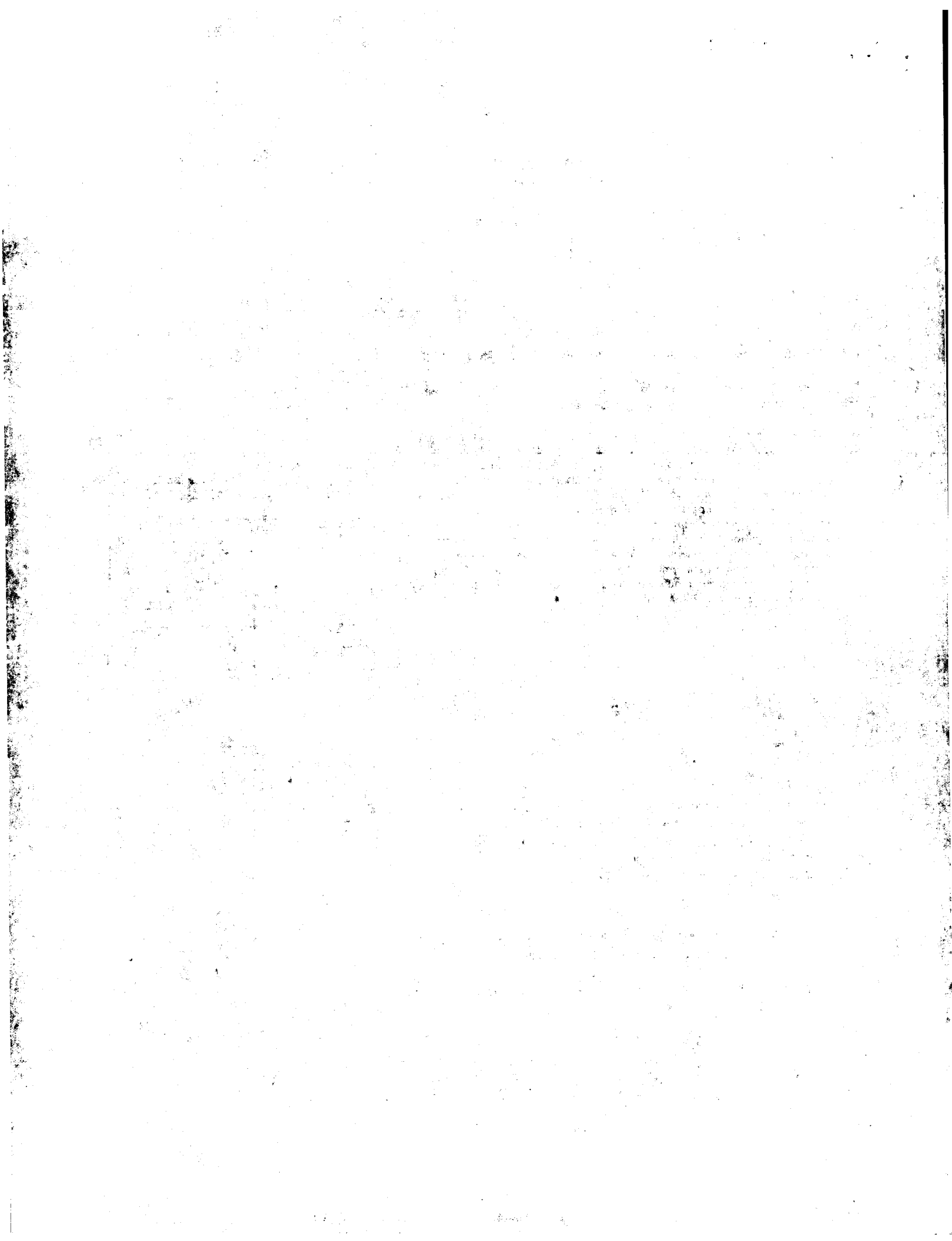
[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office



(書誌+要約+請求の範囲)

(19)【発行国】日本国特許庁(JP)

(12)【公報種別】公開特許公報(A)

5 (11)【公開番号】特開平11-65845

(43)【公開日】平成11年(1999)3月9日

(54)【発明の名称】ストール検出表示装置及び方法

(51)【国際特許分類第6版】

G06F 9/38 380

10 11/28

【F】

G06F 9/38 380 C

11/28 P

【審査請求】未請求

15 【請求項の数】19

【出願形態】OL

【全頁数】14

(21)【出願番号】特願平9-231331

(22)【出願日】平成9年(1997)8月27日

20 (71)【出願人】

【識別番号】000003078

【氏名又は名称】株式会社東芝

【住所又は居所】神奈川県川崎市幸区堀川町72番地

(72)【発明者】

25 【氏名】佐々木 洋子

【住所又は居所】神奈川県川崎市幸区堀川町580番1号
株式会社東芝半導体システム技術センター内

(72)【発明者】

【氏名】国松 敦

30 【住所又は居所】神奈川県川崎市幸区堀川町580番1号
株式会社東芝半導体システム技術センター内

(74)【代理人】

【弁理士】

【氏名又は名称】三好 秀和 (外3名)

35

(57)【要約】

40 【課題】この発明は、プログラムの開発に費やされる手間
ならびに時間を削減してプログラムの開発効率を向上させ
たストール検出表示装置及び方法を提供することを課題と
する。

45 【解決手段】この発明は、ソースプログラムを記述しなが
らストールの発生を検出し、ソースプログラムならびにソ
ースプログラムのパイプライン処理におけるパイプライン
イメージに検出したストールを強調表示するように構成さ
れる。

50

【特許請求の範囲】

【請求項1】マイクロプロセッサ用のソースプログラムを読

み込む読み込み手段と、前記読み込み手段によって読み
込まれたソースプログラムを解釈する解釈手段と、前記解
55 釈手段によって得られたソースプログラムの解釈結果に
基づいてソースプログラムのパイプライン処理における各
ステージの状態を把握する把握手段と、前記把握手段に
よって得られたソースプログラムのパイプライン処理にお
ける各ステージの状態に基づいてソースプログラムで発
60 生するストールを検出し、検出したストールのストール情
報を作成する検出手段と、前記把握手段により把握された
ソースプログラムのパイプライン処理における各ステー
ジの状態を示すパイプラインイメージを作成する作成手段と、
前記読み込み手段によって読み込まれたソースプログラ
65 ムならびに該ソースプログラムに対応して前記作成手段
により作成されたパイプラインイメージを表示し、前記検出
手段により作成されたストール情報に基づいて前記検出
手段により検出されたストールの発生個所を、表示したソ
ースプログラムならびにパイプラインイメージのそれぞれ
70 対応する個所に強調表示する表示手段を有することを特
徴とするストール検出表示装置。

【請求項2】前記読み込み手段によって読み込まれたソー
スプログラムにおいてストールの検出を行う範囲を指定す
る手段を具備し、該手段により指定された範囲で検出され
75 たストールの発生個所を強調表示することを特徴とする請
求項1記載のストール検出表示装置。

【請求項3】前記読み込み手段によって読み込まれたソー
スプログラムの編集、保存の管理を行う管理手段を具備し
たことを特徴とする請求項1又は2記載のストール検出表
80 示装置。

【請求項4】検出されたストールのストールサイクル分の
NOP(ノーオペレーション)命令をソースプログラムに自動
的に挿入する手段を具備したことを特徴とする請求項1、2
又は3記載のストール検出表示装置。

85 【請求項5】前記管理手段又は前記挿入手段によりソース
プログラムに入力が行われた場合は、ストールの発生に
対応した個所の強調表示を消去する手段を具備したことを
特徴とする請求項1、2、3又は4記載のストール検出表示
装置。

90 【請求項6】前記表示手段は、着色、下線あるいは字体に
より強調表示することを特徴とする請求項1、2、3、4又は
5記載のストール検出表示装置。

95 【請求項7】前記表示手段は、複数のストールが同時に発
生した場合にストールサイクルが最も長いストールを表示
することを特徴とする請求項1、2、3、4、5又は6記載の
ストール検出表示装置。

100 【請求項8】前記表示手段は、1つの命令で同じストール
サイクルの複数のストール発生源がある場合に、命令仕
様上先に現れるストール発生源に対応した個所を同様に
強調表示することを特徴とする請求項1、2、3、4、5、6又
は7記載のストール検出表示装置。

【請求項9】前記表示手段は、強調表示されたストール発
生源の強調表示とは異なる強調表示により強調表示され
ていない他のストール発生源に対応した個所を強調表示

することを特徴とする請求項8記載のストール検出表示装置。

【請求項10】 強調表示された個所あるいは他のストール発生源に対応した個所をメニュー表示すると、前記検出手段により作成されたストール情報を前記表示手段に表示させる手段を具備したことを特徴とする請求項8又は9記載のストール検出表示装置。

【請求項11】 前記表示手段は、リソースハザードにより連続して発生したストールが検出された場合は、リソースハザードの原因となるすべての命令を同様に強調表示することを特徴とする請求項1, 2, 3, 4, 5, 6, 7, 8, 9又は10記載のストール検出表示装置。

【請求項12】 前記表示手段は、分岐命令の成立回数を指定するフィールドを具備することを特徴とする請求項1, 2, 3, 4, 5, 6, 7, 8, 9, 10又は11記載のストール検出表示装置。

【請求項13】 前記解釈手段により条件分岐命令が解釈されると、条件分岐命令の成立／不成立を指定する手段を前記表示手段に表示させ、条件分岐命令を含む命令行ならびに該命令行を指示する手段を具備したことを特徴とする請求項1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11又は12記載のストール検出表示装置。

【請求項14】 前記表示手段は、条件分岐命令の成立／不成立の指定が変化した場合に、変化する前に検出されたストールに対応した個所と変化した後に検出されたストールに対応した個所をそれぞれ区別して強調表示することを特徴とする請求項1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12又は13記載のストール検出表示装置。

【請求項15】 前記表示手段は、条件分岐命令の成立により複数回実行される命令列をパイプラインイメージに対応させて複数回すべて表示することを特徴とする請求項1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13又は14記載のストール検出表示装置。

【請求項16】 予め設定されたパイプライン処理の基本ステージ数を越えるステージ数を要する1又は複数の命令が並行して実行され、基本ステージ数を越えるステージをパイプラインイメージとして表示する場合に、パイプラインイメージの基本ステージが表示された端に基本ステージが時間と共に進行する方向と直交する方向に並べて表示することを特徴とする請求項1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14又は15記載のストール検出表示装置。

【請求項17】 前記解釈手段は、ソースプログラムをシミュレーションするシミュレータにより実行された実行履歴に基づいてソースプログラムを解釈することを特徴とする請求項1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15又は16記載のストール検出表示装置。

【請求項18】 前記ソースプログラムは、アセンブリ言語又は高級言語により記述されてなることを特徴とする請求項1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16又は17記載のストール検出表示装置。

【請求項19】 マイクロプロセッサ用のソースプログラムを

読み込むステップと、前記ステップによって読み込まれたソースプログラムを解釈するステップと、前記ステップによって得られたソースプログラムの解釈結果に基づいてソースプログラムのパイプライン処理における各ステージの状態を把握するステップと、前記ステップによって得られたソースプログラムのパイプライン処理における各ステージの状態に基づいてソースプログラムで発生するストールを検出し、検出したストールのストール情報を作成するステップと、前記ステップにより把握されたソースプログラムのパイプライン処理における各ステージの状態を示すパイプラインイメージを作成するステップと、前記ステップによって読み込まれたソースプログラムならびに該ソースプログラムに対応して前記ステップにより作成されたパイプラインイメージを表示し、前記ステップにより作成されたストール情報に基づいて前記ステップにより検出されたストールの発生個所を、表示したソースプログラムならびにパイプラインイメージのそれぞれに対応する個所に強調表示するステップを有することを特徴とするストール検出表示方法。

詳細な説明

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、マイクロプロセッサ用に記述されたソースプログラムで発生するストールを検出し、検出したストールならびにそのパイプライン処理の流れを表示するストール検出表示装置及び方法に関する。

【0002】

【従来の技術】 VLIW, Super Scalerなどの複数命令を一度に処理するタイプでない、従来のマイクロプロセッサにおいては、ストール発生をチェックを手で行うことは難しい作業ではなかった。しかし、最近ではマイクロプロセッサの高機能化、高性能化にともなうVLIW, Super Scaleなどの方式を採用するものが増え、ストール発生をチェックを手で行うことが困難になってきた。

【0003】 元来、複数命令を一度に処理するような高機能、高性能のマイクロプロセッサにおいては、高級言語でソースプログラムを記述し、記述されたソースをコンパイラで最適化することによって効率のよいソースを作成するというのが理想であった。しかしながら、最も効率が要求される部分では人手でアセンブラソースを記述する必要が多々あり、この場合には、ストールの発生により効率が低下する部分をアセンブラソースから削除しなければ、アセンブラで記述する意味がなくなってしまふ。

【0004】 ストール発生によるプログラムの非効率性を把握するためであれば、リアルタイムエミュレータなどの実行履歴を表示するツールでも目的を達成することは不可能ではなかった。ストールの発生によるオブジェクトプログラムの処理効率の悪化を修正する手法として、

従来では例えば図15のフローチャートに示すようにソースプログラムを作成した後(ステップS31)、作成したソースプログラムをアセンブルしてオブジェクトプログラムを作成し(ステップS32)、作成したオブジェクトプログラムをリアルタイムエミュレータにロードし(ステップS33)、ロードされたオブジェクトプログラムをリアルタイムエミュレータで実行し(ステップS34)、実行履歴であるリアルタイムトレースの結果を解析してストールの発生状態を解析し(ステップS35)、ストールの発生状態に基づいてソースプログラムを修正し(ステップS36)、再びアセンブルするといった手順を実施していた。しかし、このような手法には、実際にソースプログラムをエミュレータで実行させる必要があること、ならびにプログラムを修正するたびに再コンパイルする必要があることから、ストール発生によるプログラムの非効率性の改善という目的に対して最適な手法とは言えなかった。

【0005】一方、従来では、ランゲージ・センシティブ・エディタ(Language Sensitive Editor)といわれる、キーワードを着色したり、閉じ括弧を記述時に対応する開き括弧を自動的に指摘するなど、ソースプログラムを構文解析することによって得られる情報をソースプログラム上に反映させるエディタが知られていた。しかし、このランゲージ・センシティブ・エディタは、ソースプログラムのストール発生を検出、解析することができず、ソースプログラムの記述時のみならず指定時にも指摘すべき個所を検出できず、さらにソースプログラムの解析結果であるパイプライン処理の流れを表示することができなかった。

【0006】

【発明が解決しようとする課題】以上説明したように、複数の命令を並行して処理するような場合に、ソースプログラムの処理効率を悪化させるストールの発生パターンは複雑となり、人手によりストールの発生を検出することは極めて困難であり、非効率であった。このため、エミュレータを使用してソースプログラムの実行履歴のトレース結果に基づいてストールの発生を検出する手法が従来から行われていた。

【0007】しかし、このような手法にあつては、エミュレータでソースプログラムを実行させるためにオブジェクトプログラムを作成する必要がある、また検出されたストールにしたがってソースプログラムを修正した後、再度オブジェクトプログラムを作成しなければならず、このような手順を繰り返して行うため手間と時間がかかり、極めて非効率であった。そこで、この発明は、上記に鑑みてなされたものであり、その目的とするところは、ストールの検出ならびに検出されたストールに基づくソースプログラムの修正を容易にして、プログラムの開発に費やされる手間ならびに時間を削減してプログラムの開発効率を向上させたストール検出表示装置及び方法を提供することにある。

【0008】

【課題を解決するための手段】上記目的を達成するために、請求項1記載の発明は、マイクロプロセッサ用のソースプログラムを読み込む読み込み手段と、前記読み込み手段によって読み込まれたソースプログラムを解釈する解釈手段と、前記解釈手段によって得られたソースプログラムの解釈結果に基づいてソースプログラムのパイプライン処理における各ステージの状態を把握する把握手段と、前記把握手段によって得られたソースプログラムのパイプライン処理における各ステージの状態に基づいてソースプログラムで発生するストールを検出し、検出したストールのストール情報を作成する検出手段と、前記把握手段により把握されたソースプログラムのパイプライン処理における各ステージの状態を示すパイプラインイメージを作成する作成手段と、前記読み込み手段によって読み込まれたソースプログラムならびに該ソースプログラムに対応して前記作成手段により作成されたパイプラインイメージを表示し、前記検出手段により作成されたストール情報に基づいて前記検出手段により検出されたストールの発生個所を、表示したソースプログラムならびにパイプラインイメージのそれぞれ対応する個所に強調表示する表示手段を有することを特徴とする。

【0009】請求項2記載の発明は、請求項1記載のストール検出表示装置において、前記読み込み手段によって読み込まれたソースプログラムにおいてストールの検出を行う範囲を指定する手段を具備し、該手段により指定された範囲で検出されたストールの発生個所を強調表示することを特徴とする。

【0010】請求項3記載の発明は、請求項1又は2記載のストール検出表示装置において、前記読み込み手段によって読み込まれたソースプログラムの編集、保存の管理を行う管理手段を具備したことを特徴とする。

【0011】請求項4記載の発明は、請求項1、2又は3記載のストール検出表示装置において、検出されたストールのストールサイクル分のNOP(ノーオペレーション)命令をソースプログラムに自動的に挿入する手段を具備したことを特徴とする。請求項5記載の発明は、請求項1、2、3又は4記載のストール検出表示装置において、前記管理手段又は前記挿入手段によりソースプログラムに入力が行われた場合は、ストールの発生に対応した個所の強調表示を消去する手段を具備したことを特徴とする。

【0012】請求項6記載の発明は、請求項1、2、3、4又は5記載のストール検出表示装置において、前記表示手段は、着色、下線あるいは字体により強調表示することを特徴とする。

【0013】請求項7記載の発明は、請求項1、2、3、4、5又は6記載のストール検出表示装置において、前記表示手段は、複数のストールが同時に発生した場合にストールサイクルが最も長いストールを表示することを特徴とする。

【0014】請求項8記載の発明は、請求項1、2、3、4、5、6又は7記載のストール検出表示装置において、

前記表示手段は、1つの命令で同じストールサイクルの複数のストール発生源がある場合に、命令仕様上先に現れるストール発生源に対応した個所を同様に強調表示することを特徴とする。

5 【0015】請求項9記載の発明は、請求項8記載のストール検出表示装置において、前記表示手段は、強調表示されたストール発生源の強調表示とは異なる強調表示により強調表示されていない他のストール発生源に対応した個所を強調表示することを特徴とする。

10 【0016】請求項10記載の発明は、請求項8又は9記載のストール検出表示装置において、強調表示された個所あるいは他のストール発生源に対応した個所をメニュー表示すると、前記検出手段により作成されたストール情報を前記表示手段に表示させる手段を具備したことを特徴とする。

15 【0017】請求項11記載の発明は、請求項1, 2, 3, 4, 5, 6, 7, 8, 9又は10記載のストール検出表示装置において、前記表示手段は、リソースハザードにより連続して発生したストールが検出された場合は、リソースハザードの原因となるすべての命令を同様に強調表示することを特徴とする。

20 【0018】請求項12記載の発明は、請求項1, 2, 3, 4, 5, 6, 7, 8, 9, 10又は11記載のストール検出表示装置において、前記表示手段は、分岐命令の成立回数を指定するフィールドを具備することを特徴とする。

25 【0019】請求項13記載の発明は、請求項1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11又は12記載のストール検出表示装置において、前記解釈手段により条件分岐命令が解釈されると、条件分岐命令の成立/不成立を指定する手段を前記表示手段に表示させ、条件分岐命令を含む命令行ならびに該命令行を指示する手段を具備したことを特徴とする。

30 【0020】請求項14記載の発明は、請求項1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12又は13記載のストール検出表示装置において、前記表示手段は、条件分岐命令の成立/不成立の指定が変化した場合に、変化する前に検出されたストールに対応した個所と変化した後検出されたストールに対応した個所をそれぞれ区別して強調表示することを特徴とする。

35 【0021】請求項15記載の発明は、請求項1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13又は14記載のストール検出表示装置において、前記表示手段は、条件分岐命令の成立により複数回実行される命令列をパイプラインイメージに対応させて複数回すべて表示することを特徴とする。

40 【0022】請求項16記載の発明は、請求項1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14又は15記載のストール検出表示装置において、予め設定されたパイプライン処理の基本ステージ数を越えるステージ数を要する1又は複数の命令が並行して実行され、基本ステージ数を越えるステージをパイプライン

イメージとして表示する場合に、パイプラインイメージの基本ステージが表示された端に基本ステージが時間と共に進行する方向と直交する方向に並べて表示することを特徴とする。

55 【0023】請求項17記載の発明は、請求項1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15又は16記載のストール検出表示装置において、前記解釈手段は、ソースプログラムをシミュレーションするシミュレータにより実行された実行履歴に基づいてソースプログラムを解釈することを特徴とする。

60 【0024】請求項18記載の発明は、請求項1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16又は17記載のストール検出表示装置において、前記ソースプログラムは、アセンブリ言語又は高級言語により記述されてなることを特徴とする。

65 【0025】請求項19記載の発明は、マイクロプロセッサ用のソースプログラムを読み込むステップと、前記ステップによって読み込まれたソースプログラムを解釈するステップと、前記ステップによって得られたソースプログラムの解釈結果に基づいてソースプログラムのパイプライン処理における各ステージの状態を把握するステップと、前記ステップによって得られたソースプログラムのパイプライン処理における各ステージの状態に基づいてソースプログラムで発生するストールを検出し、検出したストールのストール情報を作成するステップと、前記ステップにより把握されたソースプログラムのパイプライン処理における各ステージの状態を示すパイプラインイメージを作成するステップと、前記ステップによって読み込まれたソースプログラムならびに該ソースプログラムに対応して前記ステップにより作成されたパイプラインイメージを表示し、前記ステップにより作成されたストール情報に基づいて前記ステップにより検出されたストールの発生個所を、表示したソースプログラムならびにパイプラインイメージのそれぞれ対応する個所に強調表示するステップを有することを特徴とする。

80 【0026】

【発明の実施の形態】以下、図面を用いてこの発明の一実施形態を説明する。

90 【0027】図1は請求項1, 2, 3又は12記載の発明の一実施形態に係るストール検出表示装置の構成を示す図である。

95 【0028】図1において、この実施形態のストール検出表示装置は、GUI部1及び内部処理部2を有して構成され、アセンブラソースを作成するエディタにおいて、記述されたアセンブラソースを解析することによりアセンブラソース中で発生するストールを検出し、ストール発生原因をアセンブラソース及びアセンブラソースの解析結果であるパイプラインイメージの対応する個所に強調表示することを特徴とする。

【0029】GUI部1はユーザインタフェースの制御を行う部分で、ファイル制御部3、チェック範囲取得部4、着色情報作成部5、イメージ表示部6及びストール

情報着色部7を備えている。ファイル制御部3は、ソースプログラムのファイルの読み込み、保存、編集などの管理を行う。チェック範囲取得部4は、ストールチェック範囲を取得する。チェック範囲に応じてファイル制御部3からテキスト文字列を取得して内部処理部2に提供する。着色情報作成部5は、内部処理部2から取得したストール情報に基づいてテキスト及びパイプラインイメージへの着色情報を作成する。イメージ表示部6は、ソースプログラム及び内部処理部2から取得したパイプラインイメージ情報を表示する。ストール情報着色部7は、着色情報を使用してテキスト及びイメージ表示部6へ着色する。

【0030】内部処理部2はストールの検出を行う部分で、命令／レジスタテーブル8、テキスト解釈部9、ステージ情報作成部10、ストール情報作成部11、イメージ情報作成部12を備えている。命令／レジスタテーブル8は、VUの命令、レジスタなどの情報テーブルがあり、テキスト解釈部9で参照される。テキスト解釈部9は、GUI部1から得た1行分のテキスト文字列を、命令／レジスタテーブル8を参照して、ステージ情報作成部10に必要な情報に分解する。ステージ情報作成部10は、テキスト解釈部9から得た1行分の情報によりパイプラインの各ステージ状態を把握する。ストール情報作成部11は、ステージ情報作成部10で作成されたパイプラインのステージ状態によりストール発生状況をチェックする。ストールが発生していた場合には、ストール発生情報を作成しGUI部1へストール情報を提供する。イメージ情報作成部12は、ステージ情報作成部10で作成されたパイプラインのステージ状態により、イメージ表示部6用の情報を作成し、作成した情報をGUI部1へ提供する。

【0031】図2は上記装置における概略処理フローを示す図である。

【0032】図2において、ソースプログラムのファイルを読み込み(ステップS1)、ストールの検出範囲を取得し(ステップS2)、ソースプログラムにおけるテキスト文字列のそれぞれの命令行を解釈し(ステップS3、S4)、解釈結果に基づいてステージ情報を作成し(ステップS5)、発生しているストールを検出してストール情報を作成し(ステップS6)、パイプライン処理におけるパイプラインイメージ情報を作成し(ステップS7)、このような一連の処理をストール検出範囲におけるすべての命令行に対して行った後(ステップS8)、ストール着色情報を作成し(ステップS9)、パイプラインイメージ情報を表示し(ステップS10)、ストール情報を着色する(ステップS11)。図2において、13はファイルの読み込み処理、14はファイルの解析およびストールの検出処理、15はストール情報のファイルへの強調表示処理である。

【0033】図3は上記装置の基本画面を示す図である。

【0034】図3において、16はファイルを読み込むためのボタン、17はファイルを保存するためのボタン、

18はストールの検出を開始するためのボタン、19は本実施例の各種の初期設定を行うためのボタン、20はエラーなどのメッセージ画面を開くためのボタン、21は読み込まれているファイルの名前を表示するフィールド、22は読み込まれているファイルの行番号を表示させるためのフィールド、23は条件分岐の成立回数を指定するためのフィールド、24は読み込まれたファイル内容を表示するテキストという構成になっている。

【0035】また、本実施形態においては、2ウェイのVLIW命令形式を採用しているマイクロプロセッサを対象にしているため、読み込まれたファイル内容は、命令コードが25、26の2つの部分に分かれて表示されている。以下、本実施例では25をアッパー命令部、26をロウアー命令部と呼ぶことにする。

【0036】次に、図2のフローチャートにしたがって、本装置の動作を説明すると、まず16のボタンをクリックすると、ファイルを読み込むためのコントロールパネルが出現する。このパネルにおいてファイルを選択するか、もしくはファイル名を記述することにより、読み込むためのファイルを指定することができる。ファイルが読み込まれると、21にファイル名、22に行番号、24にファイル内容が表示される。

【0037】次に、18のボタンをクリックすると、ストールチェックを行う範囲を指定するためのメニューが表示される。ここで全ファイルを選択すると、ファイルの最初から最後までを対象にストールチェックが行われる。ストールチェックの結果、ストールが発生している場合は、24のファイル内容を表示するテキスト上のストール発生原因に着色がなされる。この着色を行った状態を図4に示す。図4に示す表示画面において、27の部分に着色がなされており、2行目のVF10と3行目のVF10においてストールが発生していることを表している。

【0038】図5はファイルの内容に加えて、表示されたファイルの内容に対応したパイプライン処理の流れを示すパイプラインイメージが表示された表示画面の様子を示す図である。図5において、28はストールチェックのために読み込まれたファイルの行番号、29はファイルの行番号28に対応するファイルの内容、30はファイルの内容29に対応するパイプラインイメージを示している。

【0039】図5に示すパイプラインイメージ30では、6ステージを持つ基本パイプラインの命令のみを使用しているソースファイルを読み込んでストールチェックを行っているため、パイプラインイメージ30は、31がMステージ、32がTステージ、33がXステージ、34がYステージ、35がZステージ、36がSステージというように6種類のステージが横に並んで表示されている。さらに、各ステージ毎に左側がアッパー命令部38、右側がロウアー命令部39となっている。また、6種類のステージ中、ストールを発生するステージは、Mステージ31とTステージ32に限られており、ストー

ルが発生している場合にはM、Tステージ31、32とも“s”が付けられて表示されている。また、発生しているストールに対してファイルの内容29における対応箇所として40で示す部分、及びパイプラインイメージ30における対応箇所として41で示す部分に同じ色で着色が行われて強調表示されている。

【0040】なお、上記実施形態において、検出されたストールのストールサイクル分のNOP（ノーオペレーション）命令をソースプログラムに自動的に挿入する手段を具備し、又はソースプログラムに入力が行われた場合は、ストールの発生に対応した個所の強調表示を消去する手段を具備するようにしてもよく、また解釈手段は、ソースプログラムをシミュレーションするシミュレータにより実行された実行履歴に基づいてソースプログラムを解釈するようにしてもよい。さらに、着色以外に下線あるいは字体により強調表示するようにしてもよい。

【0041】図6及び図7は請求項7記載の発明の一実施形態に係るストール検出表示装置の表示画面を示す図である。

【0042】図6及び図7に示す実施形態の特徴とするところは、同時に複数のストールが発生した場合に、ストールが解消されるまでに要するクロック数（以下ストールサイクルと呼ぶ）が長い方のみを、ストール発生情報として対応する個所に強調表示するようにしたことにある。図6の表示画面は、読み込まれているソースファイルにおいてストールチェックを行った結果が示されており、ここでは42で示す部分のみストール発生箇所として着色されている。

【0043】しかし、図6に対応したパイプラインイメージを加えた表示画面を示す図7においては、図6の42に対応したストール43だけでなく、同時に44で示すストールも発生している。この44のストールに対して着色が行われないのは、同時に発生している43のストールの方がストールサイクルが長いためである。具体的には、44のストールサイクルが3行目の命令がSステージに達するまでの3サイクルで、43のストールサイクルが2行目の命令がFステージに達するまでの5サイクルとなり、43のストールサイクルの方が44のストールサイクルより2サイクル長くなっている。この結果、図6、図7ともに42、43で示す部分のみストール発生情報として着色されている。

【0044】図8は請求項8記載の発明の一実施形態に係るストール検出表示装置の表示画面を示す図である。

【0045】図8に示す実施形態の特徴とするところは、1命令に対して同じストールサイクルの複数のストール発生源がある場合、命令仕様上先に現れる発生源のみ、ストール発生情報として対応する個所に着色するようにしたことにある。図8の表示画面は読み込まれているソースファイルにおいてストールチェックを行った結果を示している。ここでは、45で示す部分のみストール発生箇所として着色されている。46で示す部分も同時に同じストールサイクルのストールが発生しているが、A

DD（加算）命令に対して2つのストール発生源が存在することになり、命令仕様上先に現れるVF10のみストール発生情報として着色され、VF20については着色されないようになっている。

【0046】図9は請求項8記載の発明の他の実施形態に係るストール検出表示装置の表示画面を示す図である。

【0047】図9に示す実施形態の特徴とするところは、1命令に対して同じストールサイクルの複数のストール発生源がある場合に、命令仕様上先に現れる発生源と同じ色にパイプラインイメージのストール発生ステージも着色するようにしたことにある。図9に示す画面表示は、読み込まれているソースファイルにおいてストールチェックを行った結果のソースファイル及びパイプラインイメージを示している。図9において、47で示す2行目のVF10と3行目のVF10、48で示す2行目のVF20と3行目のVF20の2つのストールが同時に発生している。この結果、3行目のアップー命令部、ロウアー命令部それぞれにおいて2つあるソースレジスタがそれぞれ異なる色に着色されている。このような場合には、命令仕様上先に現れるレジスタへの着色と同じ色を使用して、対応するストールが発生しているパイプラインのステージへの着色を行うことにしており、アップー命令部に対応するステージである49で示すVF10と、ロウアー命令部に対応するステージである50で示すVF20と同色で着色されている。

【0048】なお、上記実施形態において、表示手段は、強調表示されたストール発生源の強調表示とは異なる強調表示により強調表示されていない他のストール発生源に対応した個所を強調表示するようにしてもよく、また、強調表示された個所あるいは他のストール発生源に対応した個所をメニュー表示すると、前記検出手段により作成されたストール情報を前記表示手段に表示させる手段を具備するようにしてもよい。

【0049】図10は請求項11記載の発明の他の実施形態に係るストール検出表示装置の表示画面を示す図である。

【0050】図10に示す実施形態の特徴とするところは、リソースハザードによるストールが連続して発生した場合に、最初にリソースハザードの原因となった命令から最後にリソースハザードの原因となった命令までを全て同色で着色するようにしたことにある。図10に示す画面表示は、読み込まれているソースファイルにおいてストールチェックを行った結果を示している。図10で読み込まれているファイル中、リソースハザードを起こす可能性のある命令は、DIV命令、SQRT命令、RSQRT命令、WAITQ命令である。ストールチェックを行った結果、51で示す2行目のDIV命令と3行目のSQRT命令と、52で示す3行目のSQRT命令と4行目のRSQRT命令と、53で示す4行目のRSQRT命令と5行目のWAITQ命令の3つのストールが発生している。

【0051】しかし、3行目のSQRT命令、4行目の

RSQRT命令が重なっているため、2行目のDIV命令、3行目のSQRT命令、4行目のRSQRT命令、5行目のWAITQ命令はリソースハザードの連続になる。このような場合には、全て同色で着色するようになっているので、2行目のDIV命令、3行目のSQRT命令、4行目のRSQRT命令、5行目のWAITQ命令は全て同色で着色されている。

【0052】図11は請求項16記載の発明の一実施形態に係るストール検出表示装置の表示画面を示す図である。

【0053】図11に示す実施形態の特徴とするところは、基本パイプラインよりステージ数の多い命令、基本パイプラインより多いステージに関しては、パイプラインイメージの右端にステージが進むごとにずらすことなく表示し、さらに基本パイプラインよりステージ数の多い命令が続けて処理されている場合は、パイプラインイメージの右端に処理されている命令のステージを、ステージが進むごとにずらすことなく並べて表示するようにしたことがある。

【0054】図11に示す画面表示は、読み込まれているソースファイルにおいてストールチェックを行った結果のパイプラインイメージを示しており、図11において、基本パイプラインのステージ部分は54で示され、この部分ではステージが進むごとに横方向にずらして表示するようになっている。一方、図11の2行目、10行目、17行目のDIV命令、9行目のELENG命令が基本パイプラインよりステージ数が多い命令である。このような命令は、まず2行目のDIVの基本パイプラインより多いステージに関しては、55で示すように基本パイプラインの右端にステージが進むごとにずらすことなく表示されている。同様に、10行目のDIV命令は56で示すように、17行目のDIV命令は58で示すように画面表示されている。9行目のELENG命令に関しては、ELENG命令の処理が終わる前に10行目のDIV命令の処理が行われているため、10行目のDIV命令の基本パイプラインより多い56で示す部分の右側に57で示すようにステージが進むごとにずらすずに表示されている。

【0055】図12及び図13は請求項13記載の発明の一実施形態に係るストール検出表示装置の表示画面を示す図である。

【0056】図12に示す実施形態の特徴とするところは、ソースファイルにおいて条件分岐命令出現時に、その条件分岐命令の成立／不成立の指示をユーザが行うことになっている場合に、対象となる条件分岐命令を含む行へテキストを自動スクロールして矢印で指示するようにしたことがある。図12に示す表示画面は、条件分岐命令を含むソースファイルを読み込んだ画面を示し、59で示すIBNE命令が条件分岐命令である。このソースファイルにおいてストールチェックを実行すると、図13に示すように条件分岐命令解釈時に、条件分岐命令の成立／不成立をユーザが指定するためのウィンドウ6

0がポップアップされ、条件分岐命令の存在する行が先頭にくるようテキストが自動スクロールされ、当該行に矢印61が付加されるようになっている。

【0057】なお、上記実施形態において、表示手段は、条件分岐命令の成立／不成立の指定が変化した場合に、変化する前に検出されたストールに対応した個所と変化した後に検出されたストールに対応した個所をそれぞれ区別して強調表示するようにしてもよく、また表示手段は、条件分岐命令の成立により複数回実行される命令列をパイプラインイメージに対応させて複数回すべて表示するようにしてもよい。

【0058】上記すべての実施形態において、ソースプログラムはアセンブリ言語以外の例えば高級言語で記述されたものであっても適用することができる。さらに、上記したそれぞれの実施形態を適宜組み合わせるようによい。

【0059】このように、本発明では、図14のフローチャートに示すように、ソースプログラムの作成を行い（ステップS21）、ソースファイルを上記実施形態による装置に読み込み（ステップS22）、ストールのチェックを行い（ステップS23）、そのストールチェックの結果を受けてソースを修正し（ステップS24）、再びストールチェックを行えばよいだけである。この結果、ソースプログラムのアセンブルが不要となり、オブジェクトプログラムの実行も不要となり、さらにストールの解析を手で行わなくてもよくなる。これにより、アセンブラでソースを記述しながらストールの検出が可能となり、さらに検出されたストールのソースファイルならびにパイプラインイメージへの強調表示が可能となり、アセンブラソースでの効率のよいオブジェクト作成が可能となる。

【0060】

【発明の効果】以上説明したように、この発明によれば、ソースプログラムを記述しながらストールの発生を検出し、検出したストールをソースプログラムならびにソースプログラムのパイプライン処理におけるパイプラインイメージに強調表示するようにしたので、オブジェクトプログラムを作成して実行することなくストールを検出してソースプログラムを修正することが可能となり、ストールの検出ならびに検出されたストールに基づくソースプログラムの修正を容易に行うことができる。これにより、ソースプログラムの開発の手間ならびに時間を従来に比べて格段に少なくすることができ、プログラムの開発効率を向上させることができる。

図の説明

【図面の簡単な説明】

【図1】請求項1、2、3又は12記載の発明の一実施形態に係るストール検出表示装置の構成を示す図である。

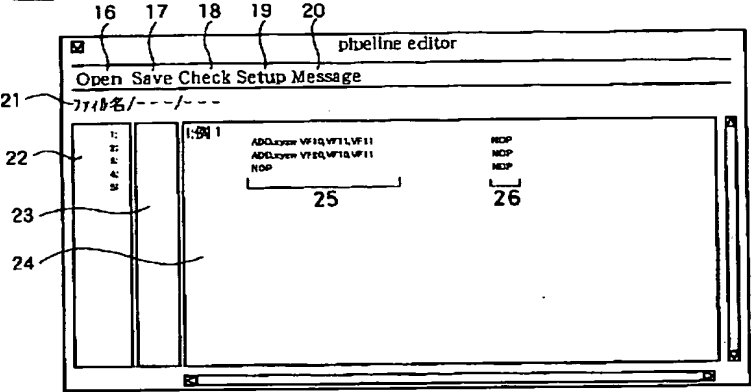
【図2】図1に示す装置における処理フローを示す図である。

- 【図3】図1に示す実施形態における基本画面を示す図である。
- 【図4】図1に示す実施形態におけるストールチェックを行った結果の表示画面を示す図である。
- 5 【図5】図1に示す実施形態におけるストールチェックを行った結果のパイプラインイメージの表示画面を示す図である。
- 【図6】請求項7記載の発明の一実施形態に係るストール検出表示装置の表示画面を示す図である。
- 10 【図7】請求項7記載の発明の一実施形態に係るストール検出表示装置の表示画面を示す図である。
- 【図8】請求項8記載の発明の一実施形態に係るストール検出表示装置の表示画面を示す図である。
- 【図9】請求項8記載の発明の他の実施形態に係るストール検出表示装置の表示画面を示す図である。
- 15 【図10】請求項11記載の発明の他の実施形態に係るストール検出表示装置の表示画面を示す図である。
- 【図11】請求項16記載の発明の一実施形態に係るストール検出表示装置の表示画面を示す図である。
- 20 【図12】請求項13記載の発明の一実施形態に係るストール検出表示装置の表示画面を示す図である。
- 【図13】請求項13記載の発明の一実施形態に係るストール検出表示装置の表示画面を示す図である。
- 【図14】請求項19記載の発明の一実施形態に係るストール検出表示方法におけるストール検出ならびに検出されたストールに基づくソースプログラムの修正の手順を示すフローチャートである。
- 25 【図15】従来におけるストール検出ならびに検出されたストールに基づくソースプログラムの修正の手順を示すフローチャートである。
- 30 【符号の説明】
- 1 GUI部 2 内部処理部 3 ファイル制御部 4 チェック範囲取得部 5 着色情報作成部 6 イメージ表示部 7 ストール情報着色部 8 命令/レジスタテーブル
- 35 9 テキスト解釈部 10 ステージ情報作成部 11 ストール情報作成部 12 イメージ情報作成部
- 13 ファイルの読み込み処理 14 テキストの解析およびストールの検出処理 15 ストール情報の強調表示処理 16 ファイル読み込み用ボタン 17 ファイル保存用ボタン 18 ストールチェック開始ボタン
- 40 19 初期設定ボタン 20 メッセージボタン 21 ファイル名フィールド 22 行番号フィールド 23 条件分岐成立回数指定フィールド
- 24 ソースを表示したテキスト 25, 38 アッパー命令 26, 39 ロウアー命令 27, 40, 42, 43, 45, 47, 48 ストール発生箇所 28 行番号 29 ソースファイル 30 パイプラインイメージ 31 Mステージ 32 Tステージ 33 Xステージ 34 Yステージ 35 Zステージ 36 Sステージ 41 ストール発生ステージ 44, 46 強調表示されないストール発生箇所 49 着色されたアッパー命令 50 着色されたロウアー命令 51, 52, 53 リソースハザードによる

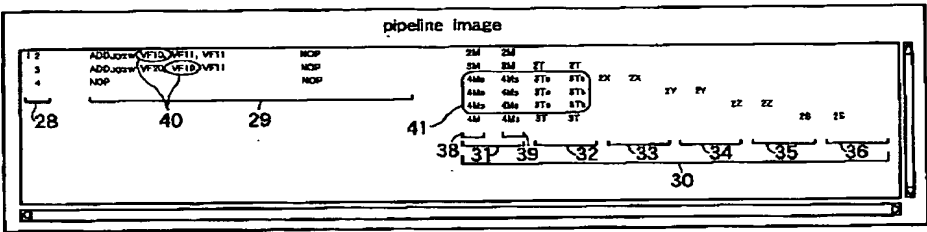
ストール発生箇所 54 基本命令のパイプラインイメージ 55, 56, 58 基本命令を超えるステージの表示 57 基本命令を超えるステージが複数ある場合の表示 59 条件分岐命令 60 条件分岐命令の成立/不成立を指定するためのウィンドウ 61 対象条件分岐命令を指示する矢印

図面

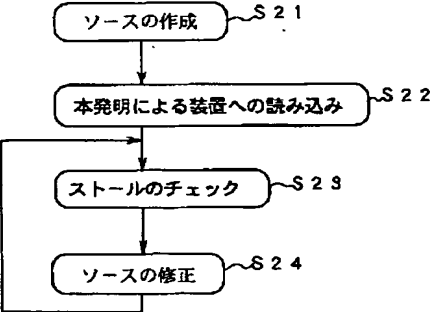
【図3】



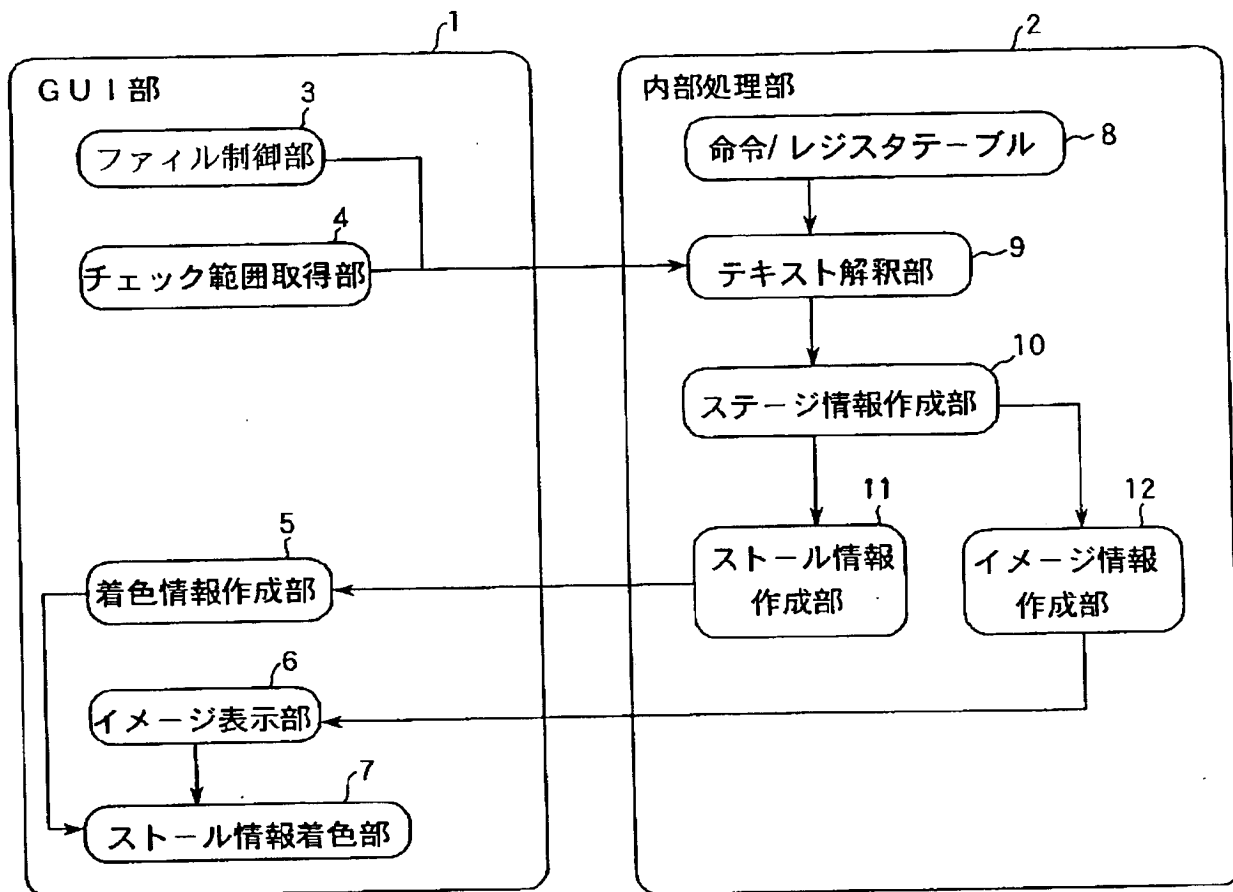
【図5】



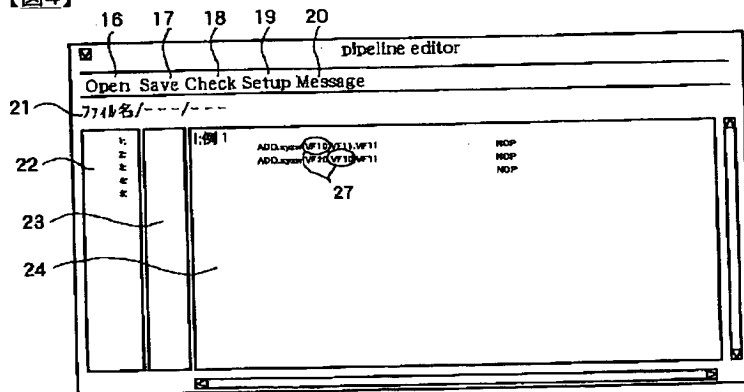
【図14】



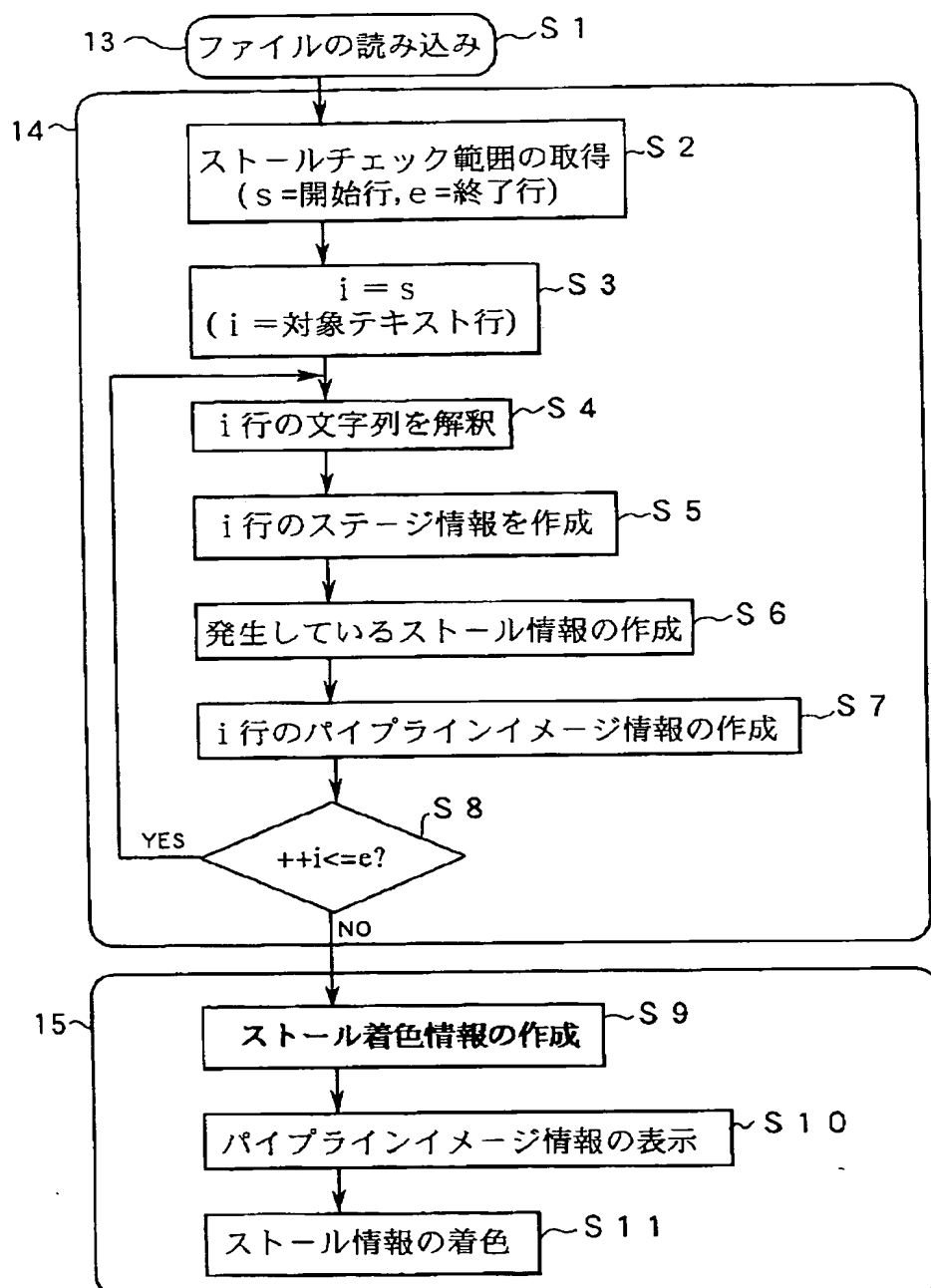
【図1】



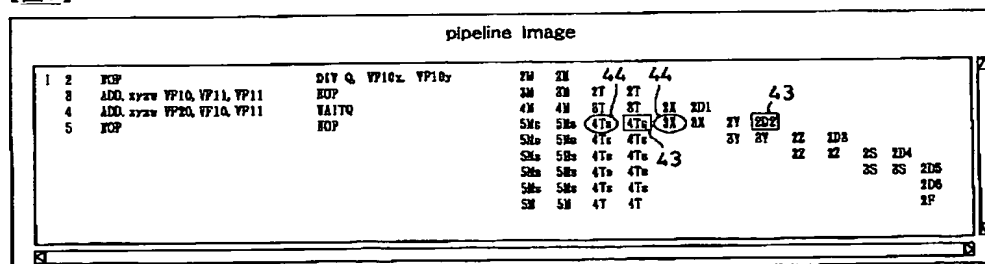
【図4】



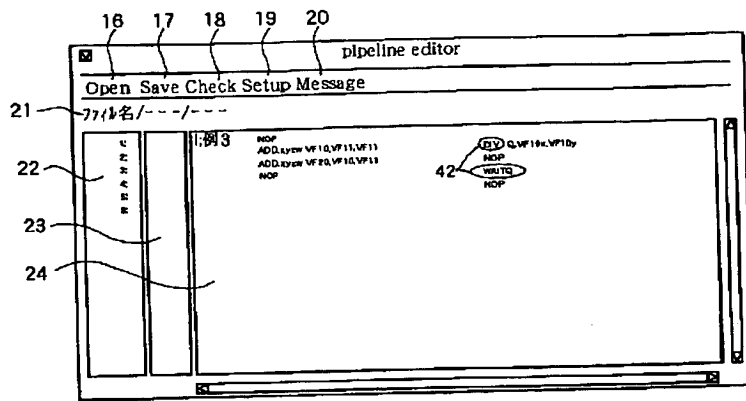
【図2】



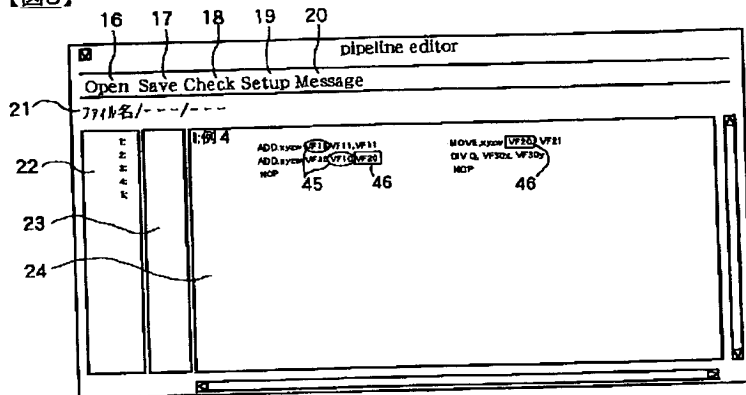
【図7】



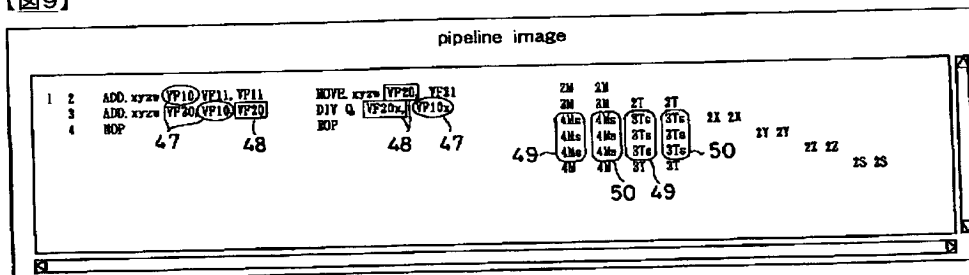
【圖6】



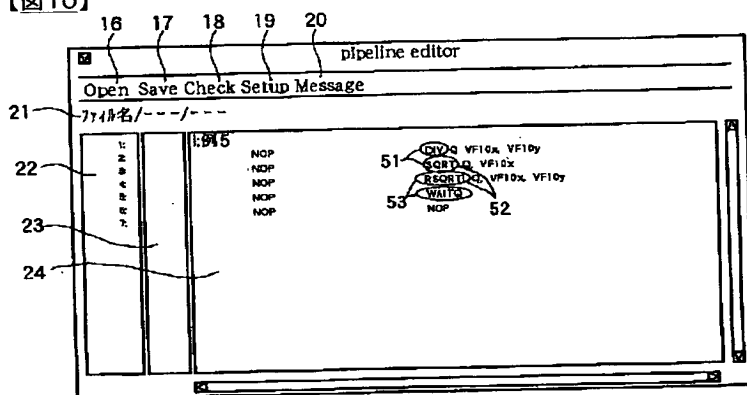
【図8】



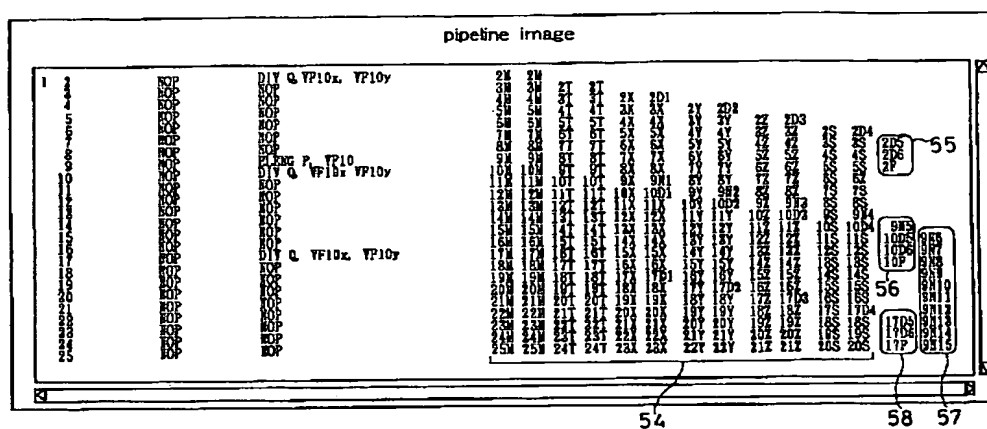
【図9】



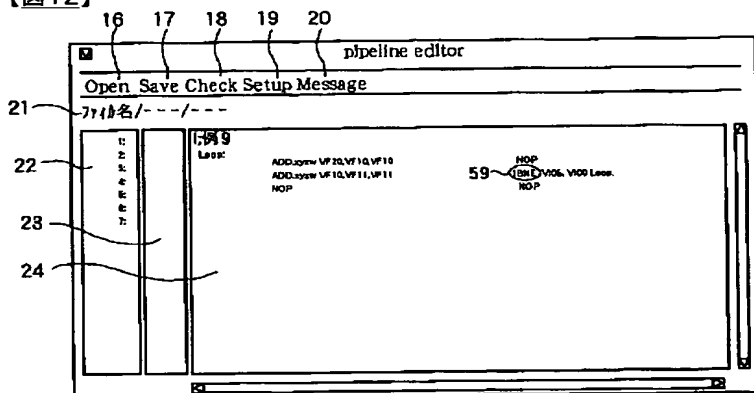
【図10】



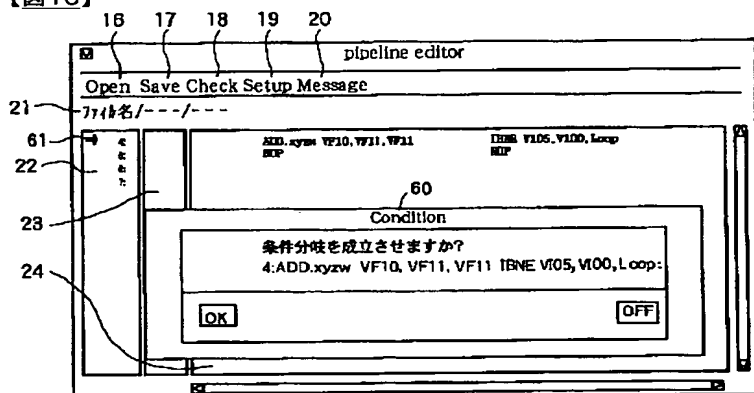
【図11】



【図12】



【図13】



【図15】

